File No. 1620/1710-03
Form A26-5707-0

IBM Systems Reference Library

IBM 1620 Input/Output Units
1621 Paper Tape Reader
1622 Card Read Punch
1624 Tape Punch
Console Typewriter

This manual describes the 1620 Input/Output Units of the 1620 Data Processing System. An introduction to paper tape and IBM card coding is included. In addition, program load routines and operation of the console typewriter are provided.

The format of the 1620 Reference Manual has been changed to conform to that of the Systems Reference Library. The original publication, A26-4500-2 and applicable newsletters have not been obsoleted.

This publication contains the Input/Output Units described in A26-4500-2. Minor changes have been made.

The Central Processing Unit is described in A26-5706.

Special Features are described in A26-5708.

Copies of this and other IBM publications can be obtained through IBM Branch Offices. Address comments concerning the content of this publication to: IBM, Product Publications Department, San Jose, California

CONTENTS

IBM 1620 Data Processing System

The paper-tape reader reads coded alphameric characters from 8-track paper tape at the rate of 150 characters per second. The characters are photoelectrically sensed, converted from 1621 paper tape code to 1620 binary-coded-decimal (BCD), and placed in core storage. If a parity error is sensed, the Read Check indicator is turned on. The computer remains in automatic mode and continues to read until the end-of-record indication (a hole in the EL channel) is reached. Whether the computer stops or continues processing depends upon the setting of the I/O Check switch. The end-of-record signal causes a record mark to be placed in core storage as the rightmost digit of the input record. NOTE: The read head area, including the lens, should be cleaned of paper dust with a lint-free cloth or tissue at least once each operating shift. Grease or oil on paper tape, from hand lotions, etc., renders it transparent, and may result in tape-read errors.

Figure 1. **1621** Paper Tape Reader

PAPER TAPE AND PAPER TAPE CODE

Data is punched and read as holes in a 1-inch-wide chad paper tape (in chad paper tape the holes are completely punched out) at a density of ten characters to the inch. An 8-track paper tape code is used. Seven positions, or tracks, across the width of the tape are used for coding numerical, alphabetic, and special characters. One track is used for EL (end-of-line) characters. Figure 2, representing a section of paper tape, illustrates the eight tracks and all coded characters.

The lower four tracks of the tape (excluding the feed holes) are used to record numerical characters in the BCD mode. For example, a hole in track 1 represents a numerical 1; a hole in track 2 represents a numerical 2; a combination of 1 and 2 punches represents a numerical 3; and so on.

The X and 0 tracks are used in combination with the numerical tracks to record alphabetic and special characters in a manner similar to zone

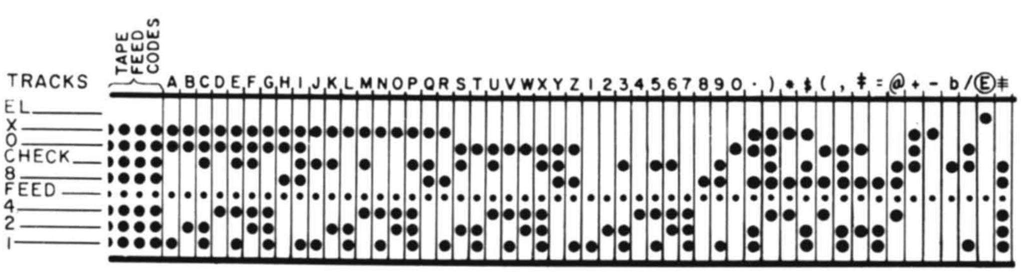


Figure 2. Paper Tape Codes

punches in IBM cards. A Read Numerically instruction causes a single X punch to read into core storage as a flag bit (negative zero).

The check track is used to establish correct parity. As a check that every character is recorded correctly, each column of the tape is punched with an odd number of holes. The EL track is not considered in the parity check.

Tape Specifications

The 1621 Paper Tape Reader and the 1624 Tape Punch are designed to operate with IBM paper tape, P/N 304469. Other paper tape of equivalent paper stock may be used, but it must conform to Electronic Industries Association specifications, RS-227.

The specifications for dimensions of punched tape can be determined after conditioning the tape to the following requirements for 24 hours:

$75^{\circ}F \pm 3.5^{\circ}$
50% RH ± 2%

1. Width of tape: 1 ± .003 inch.
2. Distance from 3-hole edge of tape to center line of feed holes: .392 ± .003 inch.
3. Vertical distance (across width of tape) between centers of holes: .100 ± .002 inch.

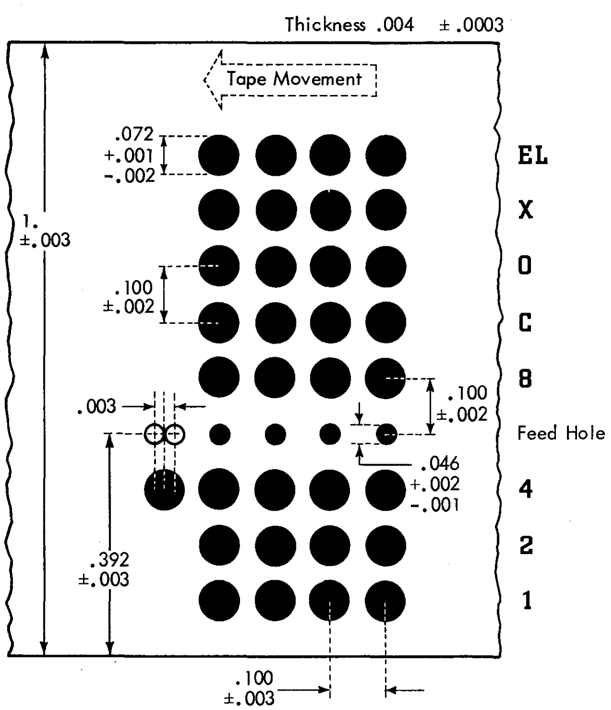


Figure 3. Paper Tape Specifications

4. Horizontal distance (parallel with edges of tape) between centers of holes:
   (a) .100 ± .003 inch for feed holes.
   (b) .100 ± .003 inch for code holes.
5. Vertical distances (across width of tape) across holes:
   (a) .072 + .001 - .002 inch for code holes.
   (b) .046 + .002 - .001 inch for feed holes.
6. Relationship of feed holes to code holes. Tolerances on the location of code holes relative to the center line of the feed hole in a row are:
   (a) ± .002 inch in a vertical direction.
   (b) ± .003 inch in a horizontal direction.
7. Thickness of tape:
   (a) Paper — .004 ± .0003 inch
8. Spliced Tape:
   Spliced tape that is within the tolerances outlined under "Punched Tape Specifications" can be used.
   (a) Total thickness of the spliced tape area must be less than 0.010 inch.
   (b) The splice must be approximately as strong as IBM paper tape.
   (c) The splice width must match the width of the tape itself.
   (d) The splice must be flexible.
   (e) The splice must not create a gum or hindrance in the tape feed area.

Tape Splicing

Paper tape handling and processing will occasionally require tape splicing when paper tape needs to be altered in length, edited, or repaired. If possible, a splice should be made in nondata portions of the tape. The ability of the tape reader to successfully and reliably read spliced tape depends upon the quality of the splice. The following is a procedure for manually splicing two lengths of paper tape together:

1. Punch tape feed codes into the two ends of the tape to be spliced together.
2. Cut the tapes at approximately a 45° angle.
3. Holding the ends of the tape with the tape feed holes, overlap the tape end in the left hand over the tape end in the right hand approximately 1/16 inch.
4. Glue in this position with holes aligned, using a quick-setting glue such as IBM tape mucilage, P/N 221030.

Other methods of tape splicing require the use of special tape splicing equipment. Appendix B contains more detailed information regarding tape splicing, including the advantages and disadvantages of types of splices.

Loading the Paper Tape Reader

Paper tape can be handled in three forms. The procedure for loading each varies slightly. The names of machine components used in the following descriptions of loading procedures are given in Figure 4.

Strip Form. Small strips of tape may be loaded directly onto the read head, as shown in Figure 5, by following this procedure:

1. Position the Reel Strip switch to STRIP.
2. Open the tape guides, form an inverted U (∩) with the leading 12 inches of paper tape, and install the tape around the read head with sufficient tension to keep the runout and tape tension contacts closed. Start on the take-up reel side of the read head. Run a finger up over the tape on top of the read head, smoothing the tape down with a firm, moderate pressure so that the tape tension bar is slightly depressed and the right side of the feed pinwheel engages the tape feed holes. Be careful not to tear the feed holes. The tape feed holes must mesh with both sides of the pinwheel.
3. Close the tape guides.

Center Roll Feed. The center roll feed eliminates the necessity for rewinding paper tape rolls to expose the starting end of the tape on the outside of the tape roll. Figure 6 shows that tape is supplied from the inside of the center roll feed, to the supply reel, around the read head, and onto the take-up reel.

The procedure for loading paper tape from the center roll feed is as follows:

1. Position the Reel Strip switch to REEL.
2. Place the reel buffer arms in the upper latched positions.
3. Open the tape guides and form an inverted U (∩) with the center section of the first eight feet of paper tape. Wrap the paper tape around the read head with sufficient tension to keep the runout and tape tension contacts closed. Start on the take-up reel side of the read head. Run a

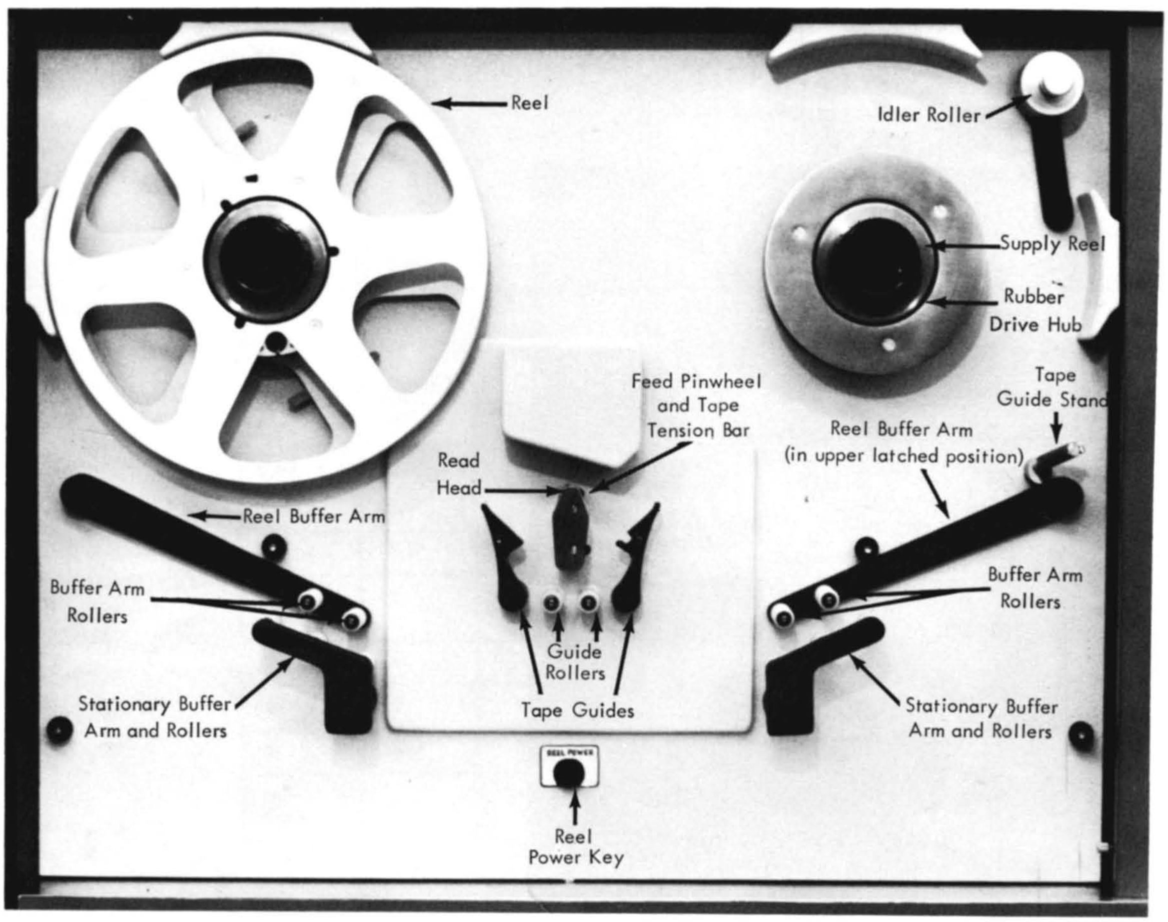


Figure 4. 1621 Tape Loading Area

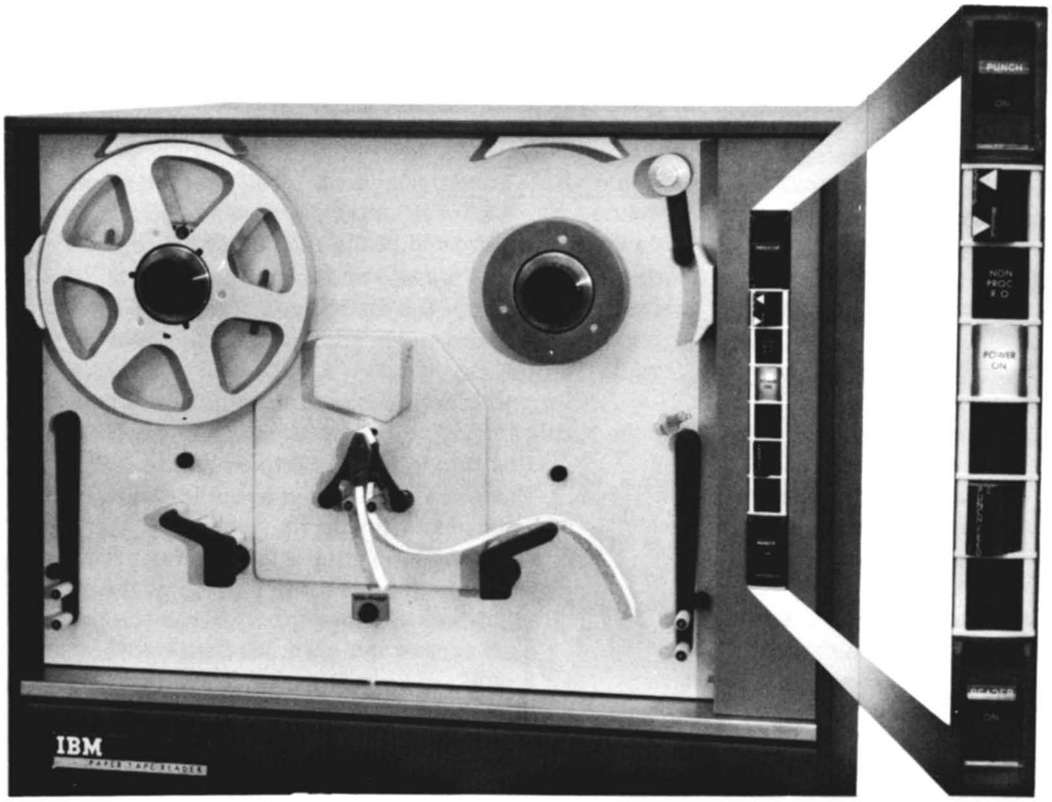


Figure 5. Strip Tape Loaded

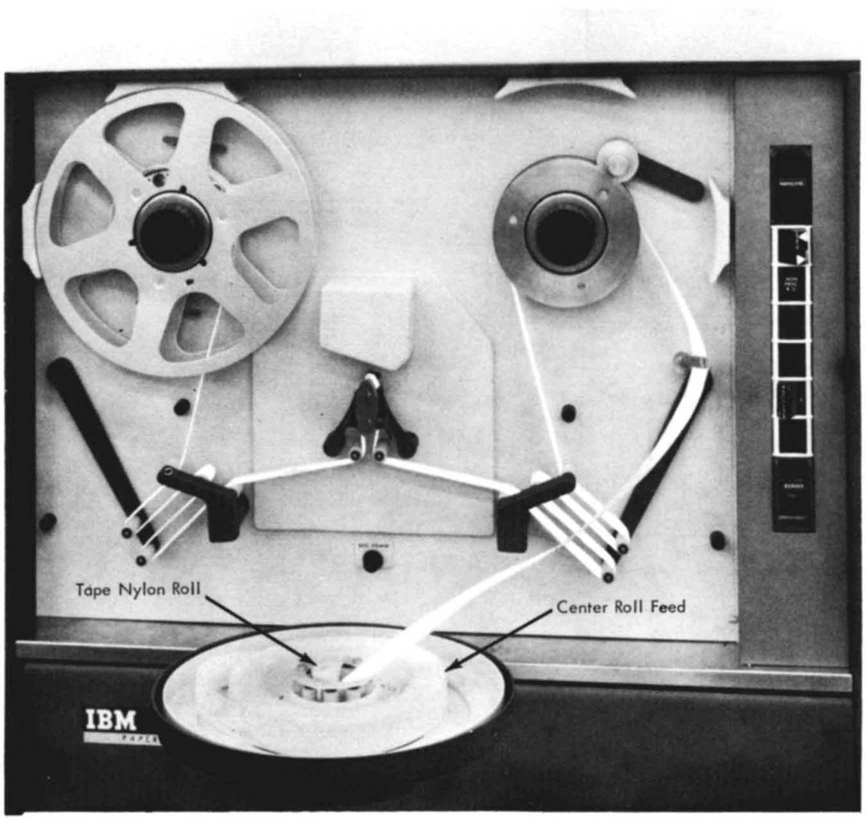


Figure 6. Center Roll Feed Loaded

finger up over the tape on top of the read head, smoothing the tape down with a firm, moderate pressure so that the tape tension bar is slightly depressed and the right side of the feed pinwheel engages the tape feed holes. Be careful not to tear the feed holes. The tape feed holes must mesh with both sides of the pin-wheel.

4. Close the tape guides.
5. Thread the leading section of paper tape under the guide roller, between the stationary buffer rollers and buffer arm rollers, and onto the take-up reel, as shown in Figure 7.
6. Thread the paper tape from the right side of the read head, under the guide roller, between the stationary buffer rollers and buffer arm rollers, over the supply reel (the rubber drive hub must be installed), around the tape guide stand, and around the tape reel nylon roll.
7. Lower the idler roller onto the supply reel.
8. Lower the buffer arms gently.
9. Depress the Reel Power key. The buffer arms should swing down to a neutral position, applying tension to the paper tape.

NOTE: The roll of paper tape must be positioned centrally, or evenly, around the center rollers to prevent excessive vibration during reading.

Reel. A reel of paper tape may be read on the 1621 by removing the rubber drive hub from the supply reel and mounting the reel of tape in its place. The tape is threaded from the right side of the reel, directly to the stationary buffer rollers, and to the take-up reel as described in the Center Roll Feed section. Figure 8 shows a reel of tape threaded on the 1621.

Operating Switches and Lights (Figure 5)

The following switches and lights are used in the operation of the 1621 Tape Reader and the 1624 Tape Punch:

Mainline Switch. With this switch on, power is supplied from the 1620 to operate the 1624 and, in

Figure 7. Threading Tape from Center Roll Feed

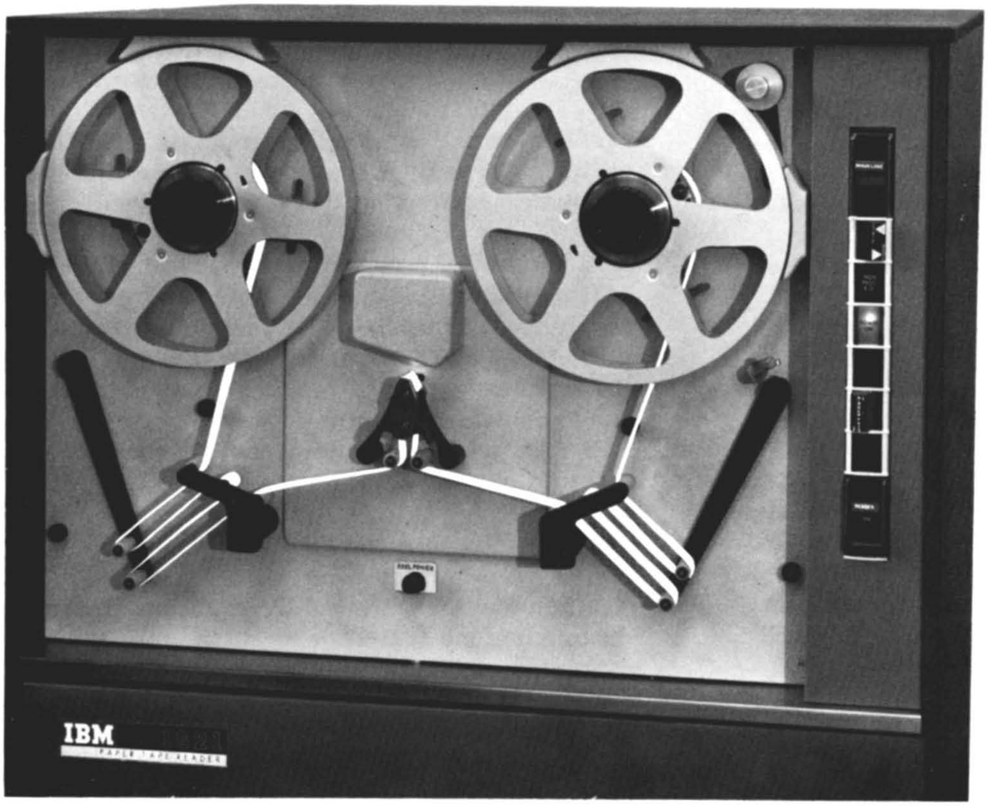


Figure 8. Paper Tape Reel Loaded

addition, to operate the 1621 with the Reader On/Off switch positioned on. The mainline switch should not be turned off with the 1620 in automatic mode because parity errors and loss of core storage data can result.

Punch Feed Switch. The 1624 punches 15 tape-feed characters per second when this switch is on.

Reader On/Off Switch. With this switch and the Mainline switch on, power is supplied for operation of the 1621.

Reel Strip Switch. In reel mode, tape is fed from the supply reel, to the left, onto the take-up reel. In strip mode, short pieces of tape may be read without reel operation.

Reel Power Key. Depress this key to operate the supply and take-up reels which position the paper tape for reading and place the machine in ready status.

Nonprocess Runout Key. Depress this key to cause paper tape to feed. Ready status is terminated and all data transfer is blocked until all paper tape has passed. Paper tape must be reloaded and the Reel Power key depressed before the machine can be returned to ready status.

Power On Light. This light is on when power is supplied from the 1620.

The tape punch is housed below the tape reader in the IBM 1621 (Figure 9) and punches data from core storage into paper tape at the rate of 15 characters per second. The characters are sent serially from core storage, starting with the location addressed by an output instruction. Each character is translated into an 8-track code before being punched.

When a record mark is sensed during the execution of a Write Numerically (WN-38) or Write Alphamerically (WA-39) instruction, an EL hole is punched and the operation stops. A Dump Numerically (DN-35) command causes punching to continue, regardless of record marks, until the highest-numbered core storage address of the 20,000-position module addressed by the DN instruction is read and punched. (See INPUT/OUTPUT INSTRUCTIONS)

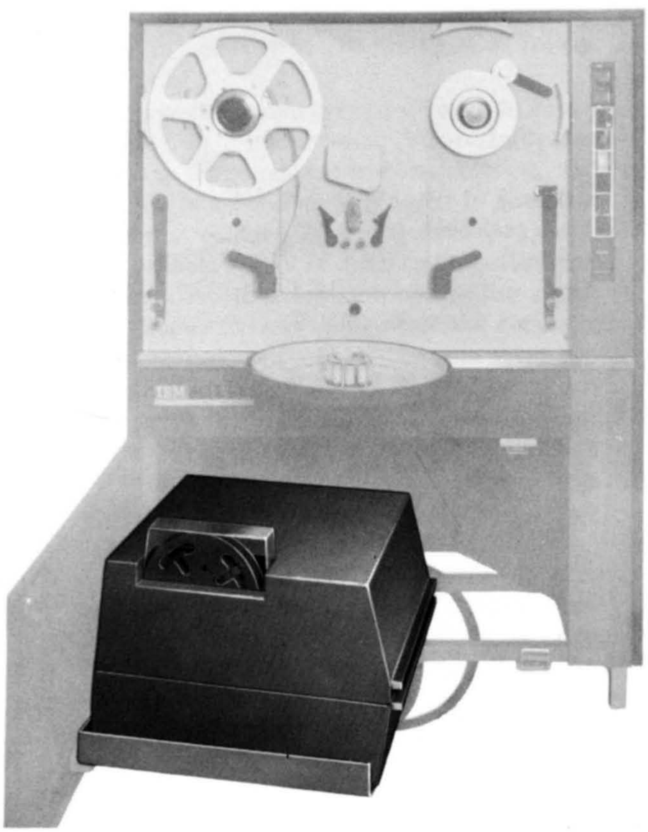


Figure 9. 1624 Tape Punch Location

At this point an EL hole is punched and the operation stops. If a character with incorrect parity is transmitted from core storage and punched, or a valid character is incorrectly punched, operation is determined by the position of the I/O Check switch on the 1620 console, as follows:

I/O Check Switch Positioned to Stop. The tape feed does not advance. The computer stops in both the automatic and manual mode; the Automatic and Manual lights and the Punch/Disk Interlock and Write Check lights on the 1620 console are turned on. Program processing can be resumed with the following "restart" procedure:

1. Position the 1624 Punch Feed switch ON.
   a. The feed code (all punches) is punched over the incorrect character.
   b. The Punch/Disk Interlock and Write Check lights are turned off.
   c. The machine is returned to manual mode only.
2. Depress the Start key on the 1620 console.
   a. The original character from storage is again punched. If an incorrect character still persists, the record may be corrected, if desired, before processing continues.
   b. The computer continues processing.

I/O Check Switch Positioned to Program. The computer branches to an error-handling routine. This routine records the fact that an error occurred, and the computer resumes operation with the incorrectly punched character in paper tape.

If the 1624 runs out of paper tape, the machine stops in automatic mode and the Punch/Disk Interlock light is turned on. Machine operation may be resumed by loading a new roll of tape (see Loading the Tape Punch) and using the "restart" procedure just described.

When the restart procedure is used to correct the incorrect punching of a valid character, and the character is again punched incorrectly, the SIE key can be used to determine the cause of the incorrect punching, as follows:

1. Use the SIE key to execute one instruction at a time.

2. When the Write Check light is turned on, observe the MBR display to determine if the character is valid.

A transient condition may cause the Write Check light (but not the Punch/Disk Interlock light) to come on even though a valid character has been correctly punched. Should this occur, briefly turn on the Punch Feed switch to turn off the Write Check light, then depress the console Start key and proceed with the program.

The punch registration (proper hole spacing) can be verified by use of a standard paper-tape gage. Off-line punching equipment can be checked in the same manner.

Loading the Tape Punch

Place the roll of unpunched tape on the turntable and thread as shown in Figure 10. The tape retainer (F) must be rotated to the left by pushing back on its extended left edge. This also moves the tape lever (D) forward to facilitate threading. An unwound section of tape is then threaded as follows:

1. Through the tape guide (A).
2. Inside the tape guide (B).
3. In front of the tape tension guide (C).
4. In back of the tape lever (D).
5. Between the punching mechanism and the punch guide block (E), which can be seen in front of the tape.
6. Between the guides on the tape retainer (F). With the end of the tape held to the left, the tape retainer (F) is returned to normal position, which causes the pins on the feed roll to pierce through the blank tape. The tape lever simultaneously returns to normal position with the top guide above the tape.

The Punch Feed switch is used to repetitively punch automatic feed punches and to provide a leader section of paper tape. The approximately 60 inches of leader needed for threading paper tape on the 1621 can be obtained from the 1624 in 40 seconds. The leader is threaded into the 1624 take-up reel so that the top edge of the tape is at the outside of the reel.

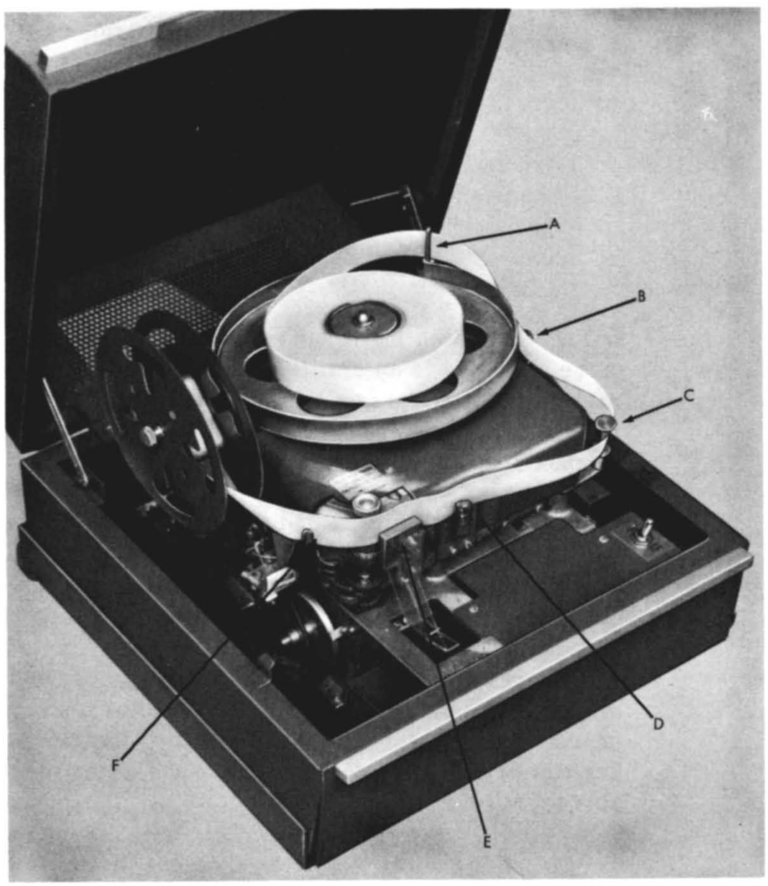


Figure 10. IBM 1624 Tape Punch

The IBM 1622 Card Read-Punch (Figure 11) provides punched card input and output for the system. The reader and punch feeds are separate and functionally independent, with individual switches, lights, checking circuits, buffer storage, and instruction codes. Under program control, up to 250 cards per minute can be read and 125 per minute punched. Reading, punching, and processing can occur simultaneously because of individual buffer storage. Buffer storage data is transferred in 3.4 milliseconds; the remaining reader and punch feed cycle time is available for processing (see Figure 12). A Last Card indicator (09) is provided for end-of-job interrogation by the programmer.

Additional advantages of card input/output are as follows:

- Compatibility of card input with other punched card equipment.
- Ease of altering input and output data without having to reproduce an entire file of data, as required with paper tape.
- Ability to read Symbolic Programming System (SPS) assembly decks directly into the 1620 without conversion to paper tape.

Figure 11. IBM 1622 Card Read-Punch

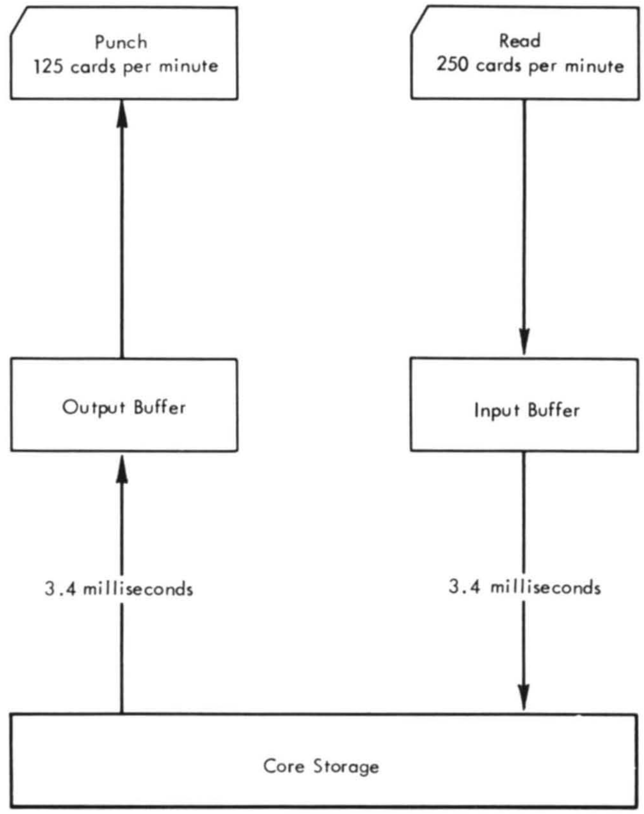


Figure 12. 1622 Buffer Storage

THE IBM CARD

The IBM card measures 7 3/8 inches by 3 1/4 inches and is .007 inches in thickness. The card stock is of controlled quality, manufactured according to rigorous specifications in order to provide strength and long life. This is necessary to ensure the accuracy of results, the proper operation of IBM data processing machines, and the continued usability of information long after it is recorded.

The card is divided into eighty vertical areas called "columns" or "card columns." They are numbered one to eighty from the left side of the card to the right. Each column is then divided horizontally into twelve punching positions. Thus the IBM card has 960 punching positions in all. The punching positions are designated, from top to bottom of the card, 12, 11 or X, 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. The punching positions for digits 0 to 9 correspond to the numbers printed on the card. The top of the card is known as the "12 edge" and the bottom as the "9 edge." These designations are made because cards are fed through machines either "9 edge first" or "12 edge first." "Face up" means the printed side is facing up; "face down," the reverse.

Each column of the card is able to accommodate a digit, a letter, or a special character. Thus the card may contain up to eighty individual pieces of information. Digits are recorded by holes punched in the digit punching area of the card from 0 to 9. For example, the card in Figure 13 shows a 1 punched in column 63, a 9 in column 72, and a 4 in column 77.

The top three punching positions of the card (12, 11 or X, and 0) are known as the zone punching area of the card. (It should be noted that the 0 punch may be either a zone punch or a digit punch.) In order to accommodate any of the 26 letters in one column, a combination of a zone punch and a digit punch is used. The various combinations of punches which represent the alphabet are based upon a logical structure or code.

The first nine letters of the alphabet, A to I, are coded by the combination of a 12 punch and digit punches 1 to 9. Letters J through R are coded by an 11 or X punch and digit punches 1 through 9. S through Z, the last eight letters, are the combination of the 0 zone punch and digit punches 2 through 9. Figure 14 illustrates alphabetic coding. The conversion of letters to and from this coding structure is done automatically by the various machines used to record or process data and it is rarely necessary to refer to data in its coded form. The eleven special characters, which are considered alphameric data, are recorded by one, two, or three punches.

Figures 13 and 14 illustrate the two most common types of corner cuts — upper left and upper right. The corner cut is used to identify visually a card type or to ensure that all of the cards in a group are facing the same direction and are right side up. Card types may also be distinguished by the use of colored cards or by a colored stripe on cards of a similar nature.

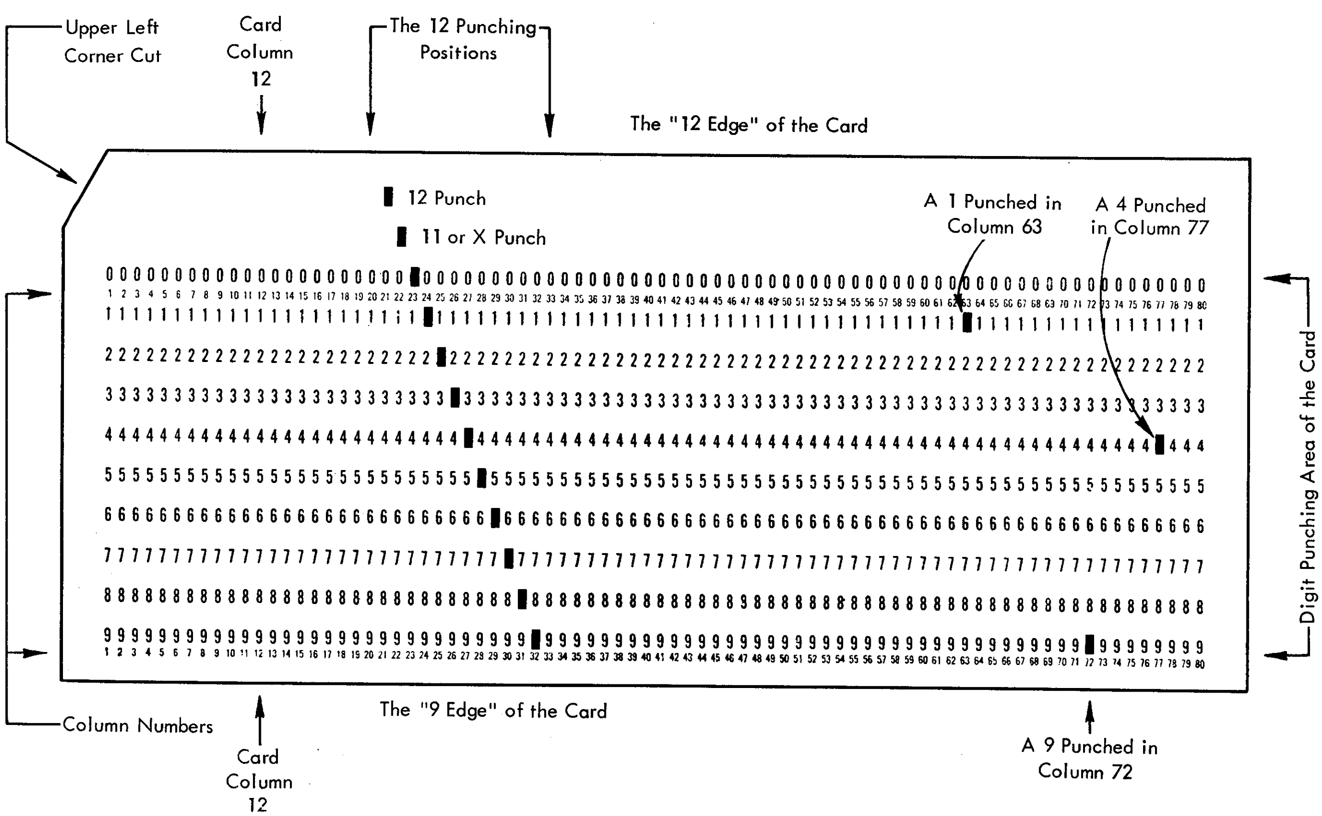


Figure 13. IBM Card

Field Definition

The fields in a card normally consist of 1 to 80 columns of data, depending on the length of the particular type of information. However, a field in 1620 core storage must consist of at least 2 digits or positions.

Numerical Data Field Definition

The high-order column of a field is punched with an 11 punch as well as the digit punch. Thus, the field defining column of a numerical field contains an alphameric character (J through R) which becomes a digit with a flag bit when read into core storage by a Read Numerically instruction (see CHARACTER CODING, Appendix A).

Alphameric Record Definition

The record-defining record mark character must be stored in core storage before or after a Read Alphamerically instruction is used to read alphameric data into core storage.

As shown in Figure 15, cards are fed from the read hopper on the right and the punch hopper on the left. Each hopper has a capacity of 1,200 cards. Both feeds have misfeeding and jam detection, and a select and a nonselect stacker. The 1,000-card-capacity stackers are of the radial type: the cards are stacked on end to permit their removal while the 1622 is running.

If either the read or punch feed is not used for approximately one minute, the drive motor for that feed is turned off to reduce noise and wear. However, the 1622 is still in ready status and responsive to a read or write command.

CARD READ

Cards are fed 9-edge first, face down, past two reading stations: check and read. Input buffer storage is initially loaded with 80 columns of card data during the Start key or Load key run-in operation. Thereafter, each card feed cycle is under program control. Data flow during card reader operation, shown in Figure 16, is as follows:

1. A read command causes a data transfer from input buffer storage to core storage. The transferred data is parity-checked in the 1620; if parity is correct, a card feed cycle follows immediately to reload buffer storage.

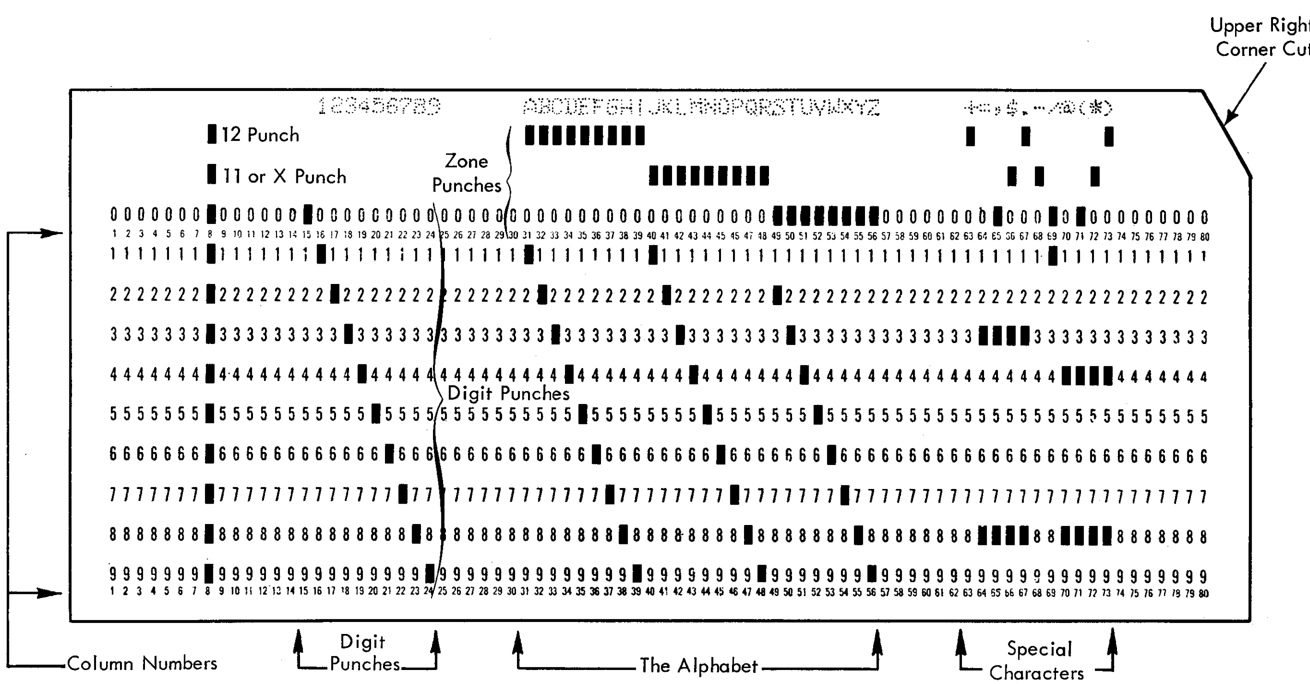


Figure 14. Card Character Coding

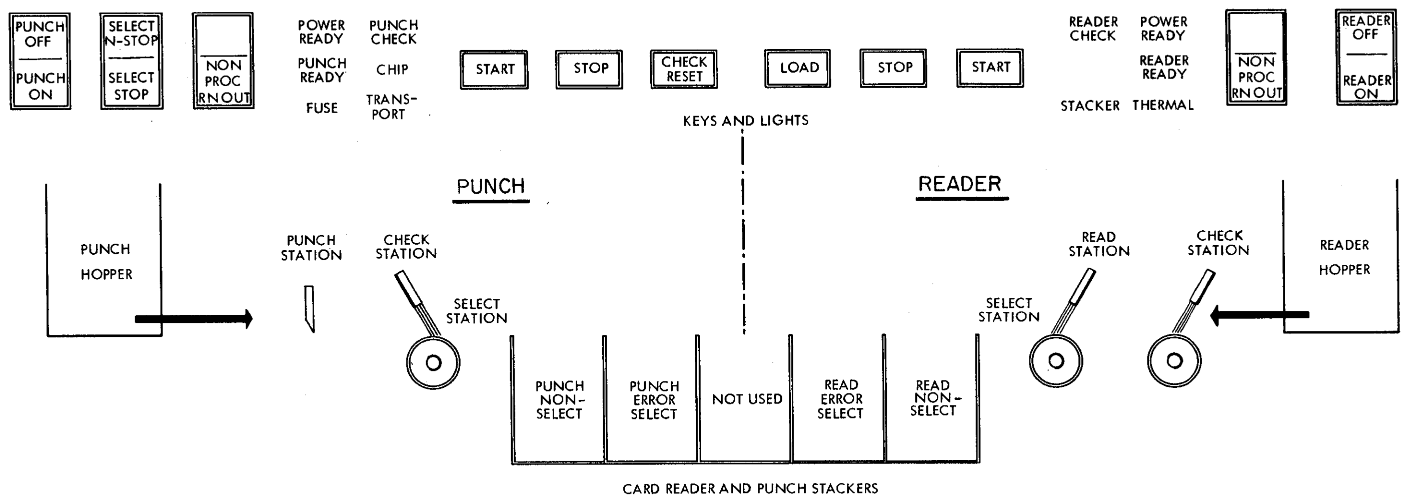


Figure 15. 1622 Keys, Lights and Card Feed

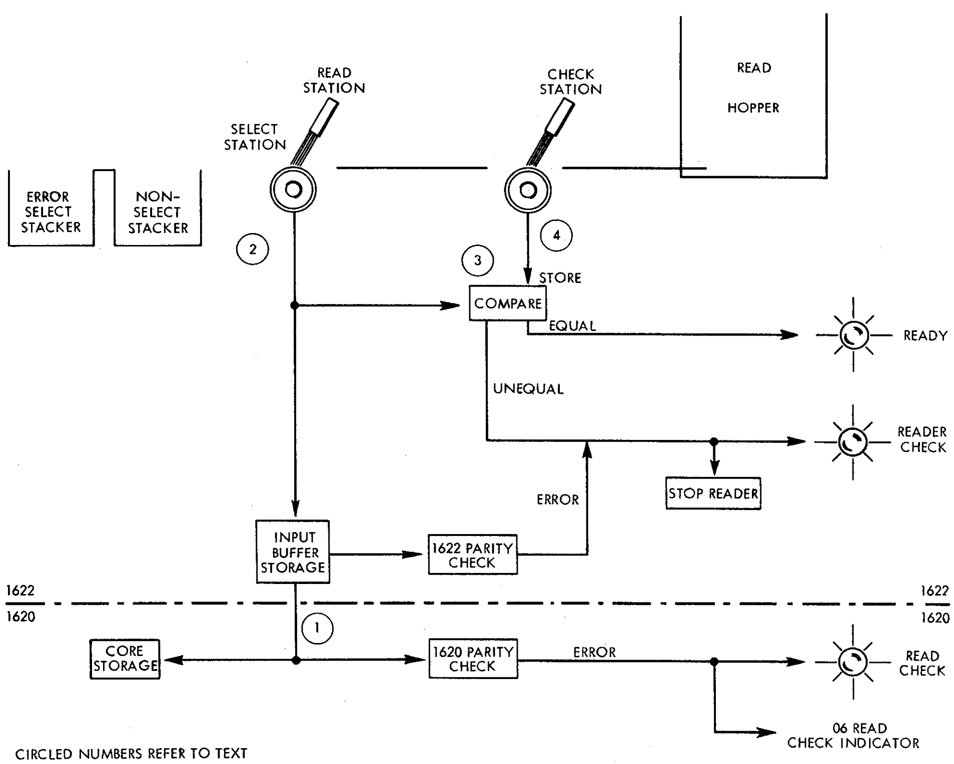


Figure 16. Read Operation Data Flow

If a parity error occurs during data transfer to the 1620, the 1620 console Read Check light and 06 indicator are turned on. The 06 indicator may be used to branch to an error-handling subroutine.

2. Following a correct transfer of data, a card is fed and new data is read into buffer storage.
3. The new data is compared at the read station against stored data previously read at the check station.

   An unequal comparison between check and read stations, or a 1622 parity error stops the reader, turns on the 1622 Reader Check light, and terminates ready status. No data transfer to the 1620 is permitted. The Reader Check light therefore cannot be on simultaneously with the 1620 Read Check light.
4. At the same time that new data is read at the read station (step 2), data on the card following is read at the check station and stored for comparison on the next card feed cycle.

CARD PUNCH

Cards are fed 12-edge first, face down, past the punch and check stations. Data flow during the card punch operation, shown in Figure 17, is as follows:

1. A write command causes a data transfer from core storage to output buffer storage. The transferred data is parity-checked in the 1620; if parity is correct, a card feed cycle follows immediately to punch the data into the card from buffer storage. The data is also parity-checked in the 1622 as it is punched into the card.

   If a parity error occurs during data transfer, the 1620 console Write Check light and 07 indicator are turned on. The 07 indicator may be used to branch to an error-handling subroutine.
2. Following a correct transfer, the data is stored for comparing (step 3), punched into the card, and parity-checked.

   If a 1622 parity error occurs, a cycle delay is initiated and the punch is

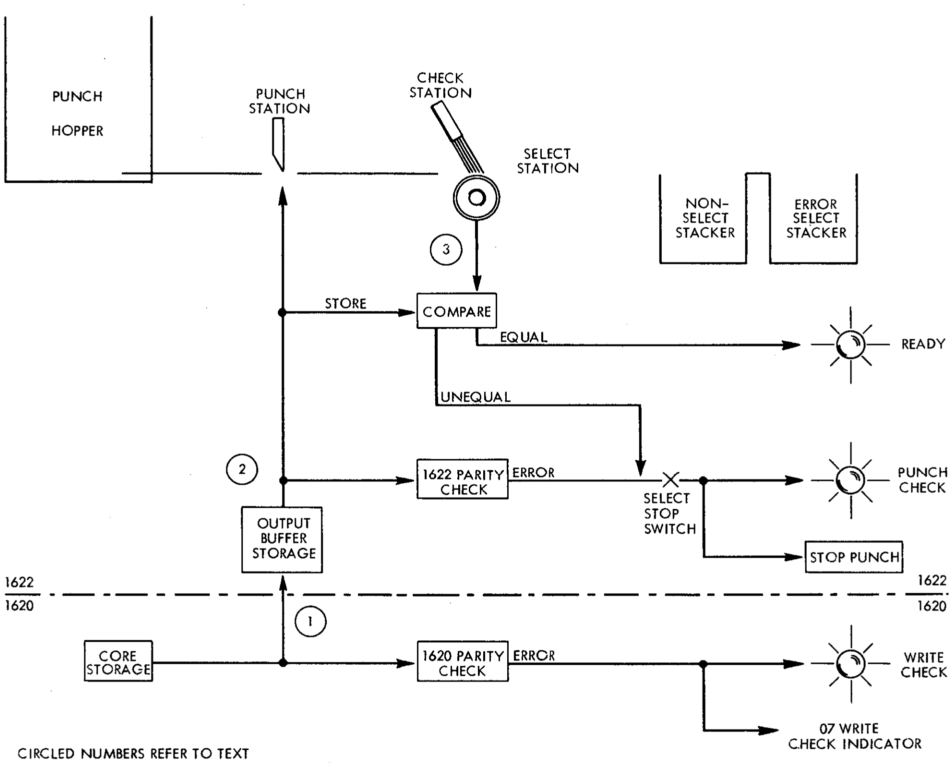


Figure 17. Punch Operation Data Flow

stopped one card feed cycle after punching the incorrect data (Select Stop switch set to STOP). The 1622 Punch Check light is turned on and ready status is terminated.

3. The card punched in step 2 is read at the check station one cycle later and compared with the data stored in step 2. An unequal comparison has the same effect as a 1622 parity error: the punch is stopped after the card cycle on which the unequal comparison occurred (no cycle delay), the Punch Check light is turned on, and ready status is terminated.

For numerical and alphameric input/output instructions, data is transferred between 1620 core storage and 1622 buffer storage in blocks of 80 or 160 digits, without consideration of record marks. A full 80 columns of card data are always transferred. Buffer storage to core storage and core storage to buffer storage transfers require 3.4 milliseconds, whether numerical or alphameric transfers are involved. When the highest-numbered core storage address of a 20,000-digit module falls within the transfer, core storage locations in the next highest 20,000 module are used or a "loop-back" to location 00000 occurs.

CARD CODING

Record marks and blank columns are processed in the following manner:

Record Marks (‡). Card columns punched 0-2-8 are read into core storage as record marks, with either a Read Numerically or a Read Alphamerically instruction. Record marks are handled as data on both input and output, and do not end the transfer of data.

A negative record mark ($\bar{‡}$), coded X-8-2, is placed in core storage as F-8-2, and punched X-8-2 in an output card.

Group Mark (ǂ). The group mark, coded C-8-4-2-1 in core storage, is used in disk storage operations to verify the correct length of records written on or read from disk storage. The group mark is recorded in disk storage as 0-8-4-2-1. The negative group mark ($\bar{ǂ}$) is recorded as C-X-O-8-4-2-1 in disk storage, as F-8-4-2-1 in core storage, and as 12-7-8 in card input/output.

Blank Card Columns. Because blank columns are read numerically into core storage as zeros, they cannot be punched in the card as blanks with a Write Numerically instruction. Therefore, cards specially punched 8-4 in all columns must be read into core storage to be used when blank columns are required in output cards. The 8-4 punches are stored as C, 8, and 4 bits, and are decoded as numerical blanks when transferred to output buffer storage.

By programming, the C, 8, and 4 bits (format blanks) are read into, or transmitted to, the output area of core storage. The output data is then transmitted into this 80-column record of blanks, leaving only the blanks required. The write instruction follows.

Double Punch Detection

Double punches are detected only if there is a duplication of BCD bits. For example, a nine (8, 1) punch and an eight (8) punch in the same column are detected as a reader check because of 8-bit duplication. However, a six (4, 2) punch and a one (1) punch in the same column are read without error as a seven (4, 2, 1) because there is no bit duplication.

BCD Coded Data

It is possible to read data recorded on IBM cards in Binary Coded Decimal form directly into the 1620 Data Processing System or the 1710 Control System from the 1622 without first transforming the data to decimal codes. This facility is described in Appendix C.

OPERATOR KEYS AND LIGHTS

The card reader and card punch have separate keys and lights (see Figures 15 and 16).

Card Reader

Reader On/Off Switch. The Reader On/Off switch is used to supply power to the reader and to turn on the Power Ready light. The 1620 Power On/Off switch must be on to make the 1622 Reader On/Off switch active.

Load Key. The Load key causes data from the first card to be checked, read into buffer storage, and automatically transferred in numerical mode to core storage positions 00000-00079. Upon completion of this data transfer, another card feed cycle occurs which loads buffer storage with data from the second card. The 1620 then simulates release and program start at 00000. The instructions from the first card, now in 00000-00079, can be used to continue loading the program or to begin processing. The 1620 must be reset and in manual mode to make the Load key operate correctly.

Start Key. The Start key is used (1) to run in cards, which are then placed under program control (data from the first card is checked and loaded in input buffer storage); (2) to set up a runout condition, which permits programmed reading of the cards remaining in the feed when the hopper has become empty; and (3) to restore ready status after the reader has been stopped by either the Stop key, an empty hopper, an error, a misfeed, or a transport jam.

Stop Key. The Stop key is used to stop the read feed at the end of the card cycle in progress and/or to remove the reader from ready status. Data that is entered into buffer storage during the read cycle in progress is transferred to core storage. The computer continues processing until the next read card command causes a reader-no-feed stop.

Nonprocess Runout Key. The Nonprocess Runout key is used to run cards out of the read feed after a reader check error, or after the Stop key has been used to stop the reader. The cards are run out into the read select stacker without a buffer storage to core storage transfer. The Reader Check light and check circuits are turned off. Cards must be removed from the hopper to make the Nonprocess Runout key active.

Reader Ready Light. The Reader Ready light is turned on to indicate that the first card has been loaded into buffer storage with the Start key, without a reader check error. It remains on until the following occurs: depression of the Stop key, a reader check error, a transport jam, a misfeed, or an empty hopper.

Reader Check Light. The Reader Check light is turned on by an unequal comparison between the read and check stations and by incorrect parity detected in buffer storage during card read. When there is an unequal comparison, the reader is stopped, ready status is terminated, and the buffer storage data just read cannot be transferred to core storage on the next read command.

1620 Console, Read Check Light. The 1620 Read Check indicator (06) and console Read Check light are turned on by a 1620 parity error during a buffer storage to core storage transfer.

1620 Console, Reader No Feed Light. The console Reader No Feed light is turned on each time the reader is selected by a read command. The light remains on, if for any reason the reader is not in ready status and the read command therefore cannot be executed. It appears to be on almost continuously when the time between read calls is less than 240 ms, indicating that processing time is available.

Card Punch

Punch On/Off Switch. The Punch On/Off switch is used to supply power to the punch and to turn on the Power Ready light. The 1620 Power On/Off switch must be on to make the 1622 Punch On/Off switch active.

Start Key. The Start key is used to feed cards to the punch station initially or after an error or nonprocess runout, and to re-establish ready status after an empty hopper, a misfeed, a transport jam, or depression of the Stop key.

Stop Key. The Stop key is used to stop the punch feed at the end of the card cycle in progress and/or to remove the punch from ready status.

Check Reset. The Check Reset key is used to reset error circuits and to turn off the Punch Check light. A Start key or Nonprocess Runout key depression follows, as described under ERROR RESTART PROCEDURES.

Select N-Stop — Select Stop Switch. This switch is used to control the stopping of the punch when error cards are selected into the punch error select stacker. With the switch set to STOP, the punch feed stops with the error card in the select stacker.

Nonprocess Runout Key. Following a punch check error, depress the Nonprocess Runout key to reset the error circuits which cause the punched card that is between the punch station and the punch check station, if it is in error, to follow the error card into the select stacker. If this card is in error, the Punch Check light is turned on again. The next two (blank) cards go into the nonselect pocket. These cards should be removed before processing is continued.

This key is also used to run out and check the last punched card of a job. Cards must be removed from the hopper to make the Nonprocess Runout key operate.

Punch Ready Light. The Punch Ready light is used to indicate that the 1622 has a card in punch position and will respond to a write command from the 1620. The Ready light is turned off by a punch check error, an empty hopper, a full chip box, a Stop key depression, a transport jam, or a misfeed.

Punch Check Light. The Punch Check light is turned on when there is an unequal comparison between the data punched and the data read (one card feed cycle later, at the check station), or when a 1622 parity error occurs during punching (Select Stop switch set to STOP). The machine stops, and ready status is terminated.

Chip Light. The Chip light is turned on to indicate that the chip box should be emptied.

1620 Console, Write Check Light. The 1620 Write Check indicator (07) and console light are turned on by a parity error during a core storage to buffer storage transfer. The 07 indicator may be used, by programming, to transfer data several times, and to halt if a correct transfer cannot be obtained.

1620 Console, Punch/Disk Interlock Light. This console light is turned on each time the punch is selected by a write command. The light remains on until the punch unit is ready and executes the command. Normally, no light is seen if commands are farther apart than 480 milliseconds. The write command cannot be executed until the punch is in ready status.

Card Reader/Punch Lights

The stacker, transport, fuse, and thermal lights are common to both the read and punch feeds and are used as follows:

Stacker Light. The Stacker light is turned on when a stacker is full. Both feeds are stopped temporarily and removed from ready status. The Ready light remains on. Operation resumes automatically after the stacker is emptied.

Transport Light. The Transport light is turned on when a card jam has occurred in either the read or punch feed or above any stacker. When this occurs, both feeds are stopped and removed from ready status. Both Start keys must be depressed to resume operation after the condition is corrected.

Fuse Light. The Fuse light turns on to indicate a blown fuse.

Thermal Light. The Thermal light is turned of if the internal temperature of the 1622 becomes excessive. After several minutes delay, the 1620 console Reset key may be depressed to turn off the Thermal light. If depression of the Reset key turns off the Thermal light, the 1620 Power switch must be turned off and then on again. Operation may be resumed after the Power Ready light has been turned on.

ERROR RESTART PROCEDURES

Reader Check Error

Cause: Unequal comparison between the read and check stations, or a buffer storage parity error. The reader stops with the error card in the select stacker (last card).

Indicators: 1622 Reader Check light ON.
1622 Ready light OFF.

Restart procedure:

1. Remove cards from the read hopper.
2. Depress the Nonprocess Runout key.
3. Remove the last three cards from the select stacker.
4. Place these three cards in front of the cards removed from the hopper and replace the deck in the hopper.
5. Depress the Start key. The card that caused the error is read into buffer storage again and if an equal comparison is obtained, the interlocked read instruction is executed and processing continues.

1620 Read Check Error

Cause: Parity error in the 1620 during data transfer from 1622 buffer storage to 1620 core storage. The reader stops with an error card (card associated with the error) in the nonselect stacker (last card). Under program control a reread can be initiated as aften as desired by the programmer.

Indicators: 1620 Read Check light ON.
1622 Reader Ready light ON.
06 Read Check indicator ON.

Restart procedure:

1. Remove cards from the read hopper.
2. Depress the Nonprocess Runout key.
3. Remove the last card from the nonselect stacker and the last two cards from the select stacker.

4. Place these three cards in front of the cards removed from the hopper. The error card from the nonselect stacker will be read in first.
5. Insert a branch to the address of the instruction that transfers the error card data from input buffer storage to core storage.
6. Depress the Start key.

Punch Check Error

Cause: Unequal comparison between the data punched and the data read (one card feed cycle later, at check station), or a 1622 parity error while punching data from buffer storage. If the Select Stop switch is set to STOP, the punch stops with the error card in the select stacker.

Indicators: 1622 Punch Check light ON.
1622 Punch Ready light OFF.

Restart procedure:
To restart without (1) immediate manual correction of the error card or (2) reprocessing of the error card.

1. Depress the Check Reset key.
2. Depress the Start key. Processing continues from the point at which the program stopped.

For manual correction of the error card:

1. Remove the last (error) card from the punch (error) select stacker and correct the error card. Place the corrected card behind those in the punch nonselect stacker.
2. Depress the Check Reset key.
3. Depress the Start key. The interlocked write command for the second card following the error card can now be executed.

For reprocessing of the error card, when one card is punched out for each card read:

1. Remove cards from both hoppers.
2. Depress both Nonprocess Runout keys.
3. Remove the last two cards from the punch error select stacker and the last two (blank) cards from the punch nonselect stacker. Also, remove the last two cards from the read nonselect stacker and the last two cards from the read select stacker.
4. Mark or destroy the two punched cards removed from the punch select stacker. Place four cards from the read stackers (nonselect in front of select) ahead of those removed from the read hopper. Place blank cards in the punch hopper.
5. Insert a branch to the address of the instruction that begins the reprocessing of the error card.
6. Depress both Start keys.

1620 Write Check Error

Cause: 1620 parity error. The error has not been punched into a card.

Indicators: 1620 Write Check light ON.
07 Write Check indicator ON.

Restart procedure:
A typeout of the core storage positions that were transferred indicates whether the data in core storage is correct. If the data is incorrect in core storage, reread the card or cards from which this data originated.

Loading procedures and a utility load routine are included in the IBM Data Processing Bulletin, Program Writing and Testing (Form J26-5547).

CONSOLE TYPEWRITER

The 1620 console typewriter (Figure 18) is used for both input and output.

Typewriter Input

The 1620 console typewriter is used to enter data and instructions directly into core storage. Off-line use is not possible because the keyboard (Figure 19) is locked except when entering data. Depressing the console Insert key (see Figure 18, I) unlocks the keyboard and permits data to be entered into core storage, starting at location 00000. Each depression of a typewriter key enters a character into core storage one location higher than the previous character. As many as 100 characters can be entered from the typewriter. After the 100th character is entered, an automatic release is initiated and the machine returns to manual mode. The Start key may then be used to start program operation at 00000.

When less than 100 characters are entered, entry of the last desired character should be followed by depressing the console Release and Start keys, or by depressing the R-S key on the typewriter keyboard. The R-S key combines the release and start functions of the console keys. The R-S symbol is typed as a permanent record that the R-S key has been used.

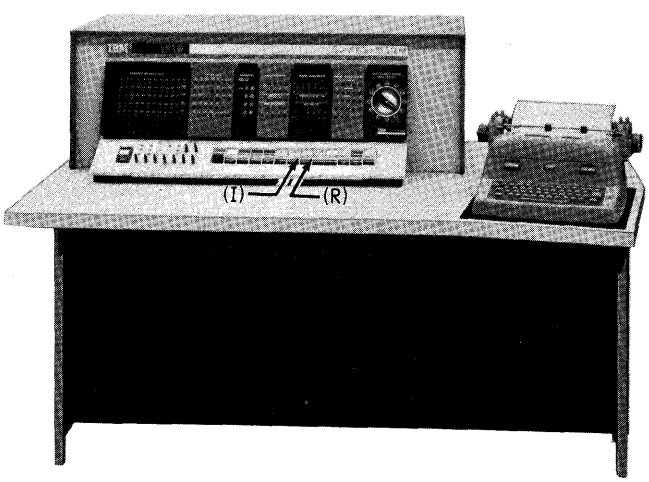


Figure 18. IBM 1620 CPU with I/O Typewriter

Programmed selection of the typewriter (Read Alphamerically or Read Numerically instructions) unlocks the keyboard, leaving the computer in automatic mode for manual entry of data on the typewriter. Data entry starts at the address-numbered location (P address) of the instruction and enters core storage at successively higher-numbered locations until the Release key is depressed.

It should be noted that the decimal point or period character in either upper shift or lower shift may be entered only with a Read Alphamerically instruction. If entry is made with a Read Numerically instruction, incorrect data is put into core storage.

If a record mark is required in core storage following the last character entered, the Record Mark key on the typewriter must be depressed before depressing either the R-S key on the typewriter or the Release key on the console (see Figure 18, R). Depressing either key again locks the keyboard and gives the computer an end-of-input/output indication.

Typewriter Output

The typewriter prints data from core storage when it is programmed to do so. When the right-hand margin is reached, the carriage returns automatically, and typing continues until a record mark is sensed or until the Release key is depressed. The Release key may be used, for example, to terminate a Dump Numerically operation from the typewriter.

Parity Checking

Input data from the typewriter is parity-checked before entering core storage. Transmission of a character with incorrect parity illuminates the console Read Check light and turns on the Read Check indicator (06).

Output data from core storage is parity-checked as it is transmitted to the typewriter. Transmission of a character with incorrect parity turns on the Write Check indicator (07) and the console Write Check light. Also, a horizontal bar is overprinted across the center of the character. An invalid character with correct parity causes a special symbol character (⌘) to print.

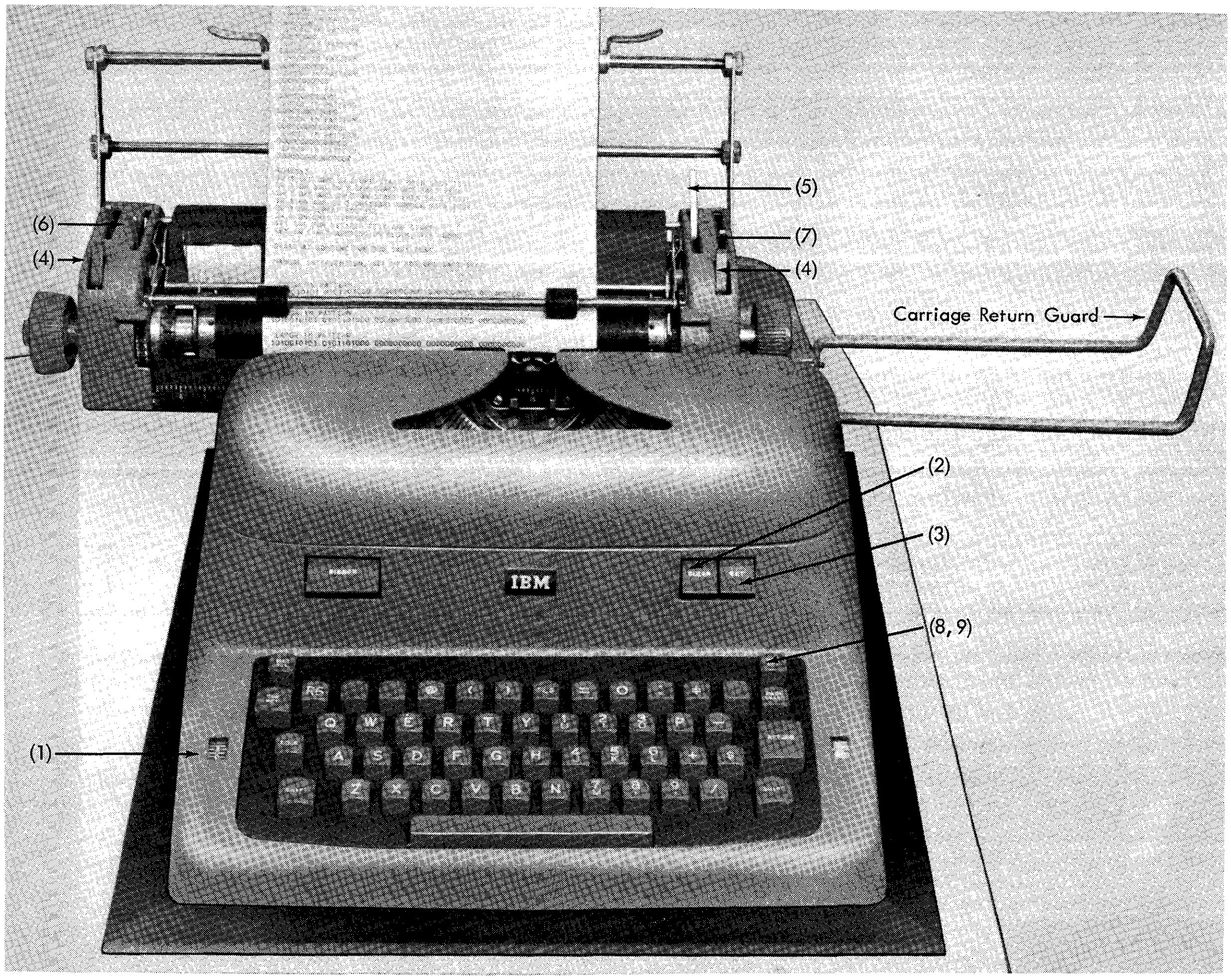


Figure 19. I/O Typewriter, Manual Controls and Keys

If a parity error occurs, the input or output operation continues until completion, and the machine either stops or continues under program control, depending on the position of the console I/O Check switch.

Manual Adjustments

The numbers preceding each of the following headings refer to numerals on Figure 19 designating particular keys, etc.; (1), for example, designates the Impression indicator.

1. Impression Indicator. The lever under this window can be positioned in settings from 0 to 10, and determines the force with which the type bars strike the paper. The higher the indicator setting, the harder the type bars strike. To test for the correct setting, move the indicator up until the comma and period print distinctly but not heavily. Use a higher setting for multiple copies, but be sure that the multiple copy lever (7) is also correctly set before finally adjusting the impression.
2. Tab Clear Lever. To clear tab stops, tabulate to the point to be cleared and depress the Tab Clear lever. To clear all stops at once, position the carriage at the right margin, hold down the Tab Clear lever, and return the carriage to the left margin stop.
3. Tab Set Lever. To set tabular stops, move the carriage to the desired position and depress the Tab Set lever. Set tab stops only when the indicator pointer is

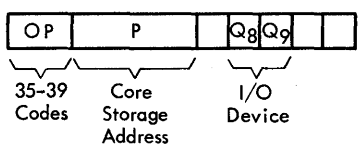
in line with a white marking on the front paper scale below it.

4. Carriage Release Lever. Depress the lever on either side to free the carriage and manually move the carriage to the right or to the left.
5. Paper Release Lever. To free the paper for positioning or quick removal, move this lever forward.
6. Line Space Lever. Moved to position 1, 2, or 3, the Line Space lever provides for single, double, or triple line spacing, respectively.
7. Multiple Copy Control. This lever moves the platen backward to compensate for the greater thickness of additional copies. As a general rule, the lever should be set at A for one to three copies and moved back one position for each additional three to five copies. Heavy print at the top of characters shows that the platen is too far back; heavy print at the bottom of characters shows that platen is too far forward. The slash symbol (/) is a good character to use in checking multiple copy settings.
8. Left-Hand Margin Set. The left margin stop is set as follows:
   a. Return the carriage to the present left margin stop.
   b. Depress and hold down the Margin Set key.
   c. Manually move the carriage as near as possible to the position desired. With the Margin Set key depressed, use the Back Space key and Space Bar to obtain the exact position desired.
   d. Release the Margin Set key.
9. Right-Hand Margin Set. The right margin stop is set as follows:
   a. Move the carriage to the left until stopped by the right margin stop.
   b. Depress and hold down the Margin Set key.
   c. Move the carriage right or left to the desired position.
   d. Release the Margin Set key.

1620 Input/Output instructions enable the transfer of data between core storage and I/O units. The Read instructions transfer data from the input unit to core storage, and the Write and Dump instructions transfer data from core storage to the output unit. The P address of the instruction determines where data is transferred to or from. The Q_8 and Q_9 digits specify which I/O unit is to be used, as follows:

01 - Typewriter
02 - Tape Punch
03 - Paper Tape Reader
04 - Card Punch
05 - Card Reader

The Q_7, Q_{10}, and Q_{11} positions of the instruction are not used.



The Read Check and Write Check indicators (codes 06 and 07), respectively, are turned on if a parity error occurs in the 1620 during input and output operations. Once the Read or Write Check indicator is turned on, it is not turned off by the reading or writing of subsequent correct characters; program interrogation or manual reset is required to turn it off.

Read Numerically (RN-36)

Description. Numerical information from an input device is transmitted serially to the P address and to successively higher-numbered core storage locations. Transmission continues until terminated by one of the following conditions:

1. Sensing of the end-of-line character when paper tape is being read. At this time a record mark character is generated automatically by the machine and placed in core storage following the last character read from tape.
2. Depression of the Release key on the console when the typewriter is used to enter information. The Release key terminates typewriter I/O operations and puts the computer in manual mode. A record mark character is not generated automatically by the machine. If it is desired to place a record mark in core storage following the last character entered, the Record Mark key on the typewriter must be depressed before depressing the Release key on the console.
3. Reading the 80th character from the card input buffer storage into core storage.

Each numerical character from an input device, along with its flag bit (if any), is stored in a single, core storage location. A parity check bit (C bit), if needed, is furnished by the machine and stored in the same location. All data that is replaced, including flag bits, is destroyed.

The — (dash) and J through R characters from the paper tape reader are entered into core storage as numerical digits with flag bits. Numerical blanks from the card reader are entered into core storage as C, 8, and 4 bits and are punched out as blanks on a Write Numerically operation. Actual blanks (from unpunched card columns) are entered into core storage as plus zeros (C bits) and are punched out as zeros on a Write Numerically operation. No other alphabetic or special characters (except the record mark) are transmitted correctly to core storage when this instruction is used.

When the typewriter is selected by a Read Numerically instruction, the 1620 stops in automatic mode to await the manual entry of information from the typewriter keyboard.

Although a Read Numerically instruction may be used to transfer alphameric data from the 1622 without a parity error occurring, several characters (equal sign, period, dollar sign, and comma) besides the record mark will have an 8, 2 representation in core storage. These characters will be treated as record marks during execution of Write Numerically and Transmit Record instructions. Thus, it behooves the programmer to be aware of card data and format before using the Read Numerically instruction to read alphameric data.

Use of the Read Alphamerically instruction eliminates this problem.

The core storage characters that result from the transfer of alphameric card data with a Read Numerically instruction are shown in Figure 20.

Execution Time. Execution times for the paper tape reader and typewriter depend upon the speed of the input device selected and the number of characters read. Execution time is 3.4 milliseconds for transferring 80 columns of data from the 1622 input buffer storage to core storage.

Read Alphamerically (RA-37)

Description. Alphameric information from an input device is transmitted serially to the P address and to successively higher-numbered core storage locations.

The units digit (P_6) of the P part of the instruction must be an odd number; otherwise, the input information is not placed in core storage correctly and parity errors may occur when the input information is read in. This is due to the 2-character transfer operation of core storage. The odd-num-

Character	Bits entered into Core Storage by Read Numerically Instruction					
	C	F	8	4	2	1
A						X
B					X	
C	X				X	X
D				X		
E	X			X		X
F	X			X	X	
G				X	X	X
H			X			
I	X		X			X
J	X	X				X
K	X	X			X	
L		X			X	X
M	X	X		X		
N		X		X		X
O		X		X	X	
P	X	X		X	X	X
Q	X	X	X			
R		X	X			X
S					X	
T	X				X	X
U				X		
V	X			X		X
W	X			X	X	
X				X	X	X
Y			X			
Z	X		X			X

Character	Bits entered into Core Storage by Read Numerically Instruction					
	C	F	8	4	2	1
0	X					
1						X
2					X	
3	X				X	X
4				X		
5	X			X		X
6	X			X	X	
7				X	X	X
8			X			
9	X		X			X
/						X
. (period)			X		X	X
, (comma)			X		X	X
@	X		X	X		
(	X		X	X		
)	X		X	X		
=			X		X	X
*		X	X	X		
-		X				
+	X					
Card I/O Only 11,0		X				
Card I/O Only 12,0	X					
‡	X		X		X	
⧧	X		X	X	X	X
$	X	X	X		X	X
Blank	X					

NOTE: Dollar sign, equal sign, period, and comma are interpreted as record marks on Write Numerically and Transmit Record Instructions.

Figure 20. Core Storage Data Resulting from Reading Alphameric Card Data with RN Instruction

bered location must contain the right-hand (numerical) digit of the 2-digit alphameric code. Transmission continues until terminated by one of the following conditions:

1. Sensing of the end-of-line character when paper tape is being read. At this time an alphameric record mark character (a numerical zero digit followed by a single record mark character) is generated automatically by the machine and placed in core storage following the last character read from tape.
2. Depression of the Release key on the console when the typewriter is used to enter information. An alphameric record mark character is not generated automatically by the machine. If it is desired to place an alphameric record mark in core storage following the last character entered, the Record Mark key on the typewriter must be depressed before the Release key on the console is depressed.
3. Reading the 80th character from card input buffer storage into the 159th and 160th positions of core storage. A record mark is not generated in storage.

Information from an input device may be a random mixture of numerical, alphabetic, and special characters. Each character from an input device is stored in core storage as two digits. Flag bits are not transmitted on characters read by an input device; however, flag bits already in the core storage area where the information is read in remain unchanged. A single record mark character read by an input device is stored in core storage as a numerical zero digit (C bit) followed by a single record mark character (coded C-8-2).

Numerical data stored in the two-digit alphabetic mode must be converted by programming to single-digit numerical data before being used in arithmetic commands. The Transfer Numerical Strip instruction (special feature) may be used for this conversion.

When the typewriter is selected, the 1620 stops in automatic mode to await the manual entry of information from the typewriter keyboard.

Execution Time. Same as Read Numerically.

Write Numerically (WN-38)

Description. Numerical information in the P address and in successively higher-numbered core storage locations is transmitted serially to an output device. Transmission continues until terminated by one of the following conditions:

1. Sensing of a record mark character in core storage. The record mark character is not written on the typewriter, but causes an end-of-line character to be punched in paper tape.
2. Depressing the Release key on the console. If the Release key is not depressed, and a record mark is not encountered before the data at the highest-numbered core storage address is written, the machine "loops back" to 00000 and transmission continues.
3. Writing the 80th position in card output buffer storage.

Each numerical character in core storage, and its flag bit (if any) is written on an output device, and the character in core storage remains unchanged. No alphameric or special character represented in core storage as two numerical characters can be written on an output device as a single character by this instruction.

The P address must not be the location of a record mark.

Execution Time. Execution times for the paper tape reader and typewriter depend on the speed of the output device selected and the number of characters written. Execution time equals 3.4 milliseconds for transferring 80 columns of data from core storage to 1622 output buffer storage.

Write Alphamerically (WA-39)

Description. Alphameric information from the P address and from successively higher-numbered core storage locations is transmitted serially to an output device. The units digit (P_6) of the P part of the instruction must be an odd number, otherwise, the information in core storage is not converted correctly to the single-character output representation. Transmission continues until terminated by one of the following conditions:

1. Sensing of the alphameric record mark. A record mark character is not written on the typewriter but causes an end-of-line character to be punched in the tape.
2. Depression of the Release key on the console. If this is done before an alphameric record mark has been encountered in core storage, a record mark character is not

written, and if the device is the paper tape punch, an end-of-line character is not punched. If the Release key is not depressed and no alphameric record mark is encountered before the data from the highest-numbered core storage address is written, the machine "loops back" to 00000, and transmission continues.

3. Writing of the 159th and 160th core storage positions into the 80th position of card output buffer storage.

Each alphameric character in core storage is written on the output device as a single character, and the character in core storage remains unchanged. No flag bit is written on the output device.

The P address must designate an odd-numbered core storage position and must not be the location of a record mark.

Execution time. Same as Write Numerically.

Dump Numerically (DN-35)

Description. Numerical information is transmitted serially to an output device, beginning with the P address and continuing through successively higher-numbered addresses. Transmission terminates after the character has been written from the highest-numbered position of the addressed core storage module. This address is 19999, 39999, or 59999, depending on the 20,000-position module specified by the P address. If the output device selected is the tape punch, an end-of-line character is punched in the tape immediately following the last character. If the output device selected is the card punch, punching continues until all 80 columns of the last card have been punched out. The instruction 35 00000 00400 causes the first 20,000 digits in core storage to be punched into 250 cards. If a starting address chosen is not an exact multiple of 80 columns to the end of a 20,000-digit storage module, data from the last card will overflow to the "low" end of the next module, or "wrap around" to address 00000 and successively higher-numbered addresses as required to completely punch out all 80 columns of the last card. Transmission also may be terminated at any time by depressing the Release key on the 1620 console.

Except for the 0 punchout on cards, each numerical character, as well as any single record mark character, is written on the output device along with its flag bit (if any). The character in core storage remains unchanged. This causes only the flag (X punch) of a 0 to be punched in an output card, so that any subsequent printout of that character from the card will be as a hyphen (see Appendix A).

An alphameric character (represented in core storage as two numerical digits) cannot be written on the output device as a single character by this instruction.

Execution Time. Same as Write Numerically.

Examples of a paper tape load routine and a card load routine are explained in detail in this section to illustrate the concept of program loading.

PAPER TAPE LOAD ROUTINE

Large records, like small records, consist of data (program instructions are also considered data) and an EL punch in paper tape. The EL punch, which terminates the record, enters core storage as a record mark (‡). The format below represents a large record in core storage.

DATA. . . . . ‡

The following format represents the same record separated into four smaller records by EL punches in the paper tape.

DATA. . ‡ DATA. . ‡ DATA. . ‡ DATA. . ‡

Each EL punch causes a record mark to replace a character in core storage. Where individual records are interspersed in core storage, rather than stored in a continuous line, it is sometimes necessary to restore the characters that the record marks replace. This is done by transferring each character to another location before the record mark replaces it, and then, after the record mark has been stored, to transfer the character back, wiping out the record mark. The following load routine saves these characters. Read the explanation completely for clarification of early steps.

The format of the load routine and of data records in paper tape is:

LOAD ‡ $\bar{d}dddd_1$36$aaaaa_1$ ‡ $DATA_1$. . ‡ $\bar{d}dddd_2$36$aaaaa_2$ ‡ $DATA_2$. . ‡ . The load routine is represented by LOAD. Each $\bar{d}$dddd36aaaaa is an addressing record for the next data record (DATA. . ‡). $\bar{d}$dddd is the address of the character that will be replaced by the following DATA record mark, and aaaaa is the address that the following DATA record will be read into. The LOAD and addressing record marks are of no concern.

The load routine (LOAD) is read in by use of the Insert (36 00000 00100), Release, and Start keys. The load routine is as follows:

00000	41 00000 00000	No OP
00012	36 00031 00300	Read paper tape into 00031
00024	25 00071 ddddd	Transfer digit from location ddddd to 00071
00036	36 aaaaa 00300	Read paper tape into aaaaa
00048	26 00066 00035	Transfer field from 00035 to 00066
00060	15 00000 00000	Transfer Digit Immediate
00072	49 00012	Branch

After the load routine reads from paper tape into core storage locations 00000-00079 as a result of the above Insert operation, Release and Start key depressions are again required to initiate computer operation at 00012. The second Load instruction is executed, and the first addressing record, $ddddd_1$36$aaaaa_1$ ‡ , replaces part of the third and fourth Load instructions. The placement of the ‡ in Q_7 of the fourth instruction is of no concern.

ddddd is the predetermined location of the character in core storage that the next DATA record mark temporarily replaces. The character is saved (third instruction) and transferred back (sixth instruction) to replace the ‡.

aaaaa is the core storage location for each DATA record.

The third instruction, now 25 00071 $\bar{d}$dddd, is executed. The character at ddddd is transferred to 00071 (Q_{11} of the sixth instruction).

The fourth instruction, now 36 aaaaa 00300, is executed. The next DATA record enters core storage, beginning at location aaaaa.

The fifth instruction, 26 00066 00035, is executed, and $\bar{d}$dddd becomes the P address of the sixth instruction.

The sixth instruction, now 15 $\bar{d}$dddd 0000X (X is the saved character) transfers the saved character back to its original location. Thus, the ‡ at the end of each data record is replaced by the character that was there to begin with.

The last instruction of the load routine, 49 00012, branches the computer to the second instruction, and reads in the addressing record for the next data record.

The last addressing and data records are used to branch the computer to the starting address of the loaded program. They are $\overline{0}$00793600074 ‡ and sssss ‡ (sssss is the starting address).

CARD LOAD ROUTINE

The following five-instruction load routine is also an example of Indirect Addressing (special feature). Without Indirect Addressing, the program would require twelve instructions. The program is straightforward and simple, and in addition simplifies the preparation of program cards. Other advantages are:

1. Program cards may be loaded in any sequence.
2. The address being loaded is punched in card columns 1 to 5 for easy card identification.
3. From one to sixty digits (as many as five instructions) may be loaded by each program load card.
4. After loading data from the last program cards, the program will branch to any address specified by the programmer in the last card and start the main program.
5. Data cards may be placed in the card reader with the program deck, thereby reducing card handling.
6. Cards containing corrections to the original program (patch cards) can be added to the program deck whenever necessary.
7. To simplify the explanation, 19901-19980 is used as a read-in area. It cannot be loaded by the program. The last paragraph describes a program that avoids this restriction.

Operating Instructions

To load the program, reset the 1620, place the card deck in the 1622 Card Read-Punch, and press the Load key.

Card Sequence

The cards are arranged in the following sequence:

1. The program loader card (one card, punched with the program below).
2. Program cards (see their format below).
3. The last program card, containing the branch address.
4. Data cards, if any.

The Program Loader Card

This card is punched in card columns 1 to 56 with the following information:

36 19901 00500 25 00080 1991$\overline{0}$ 31 1990$\overline{5}$ 19920 25 1991$\overline{0}$ 00080 49 1991$\overline{5}$ ‡. It is loaded to locations 00000 to 00079 by the Load key. Note that each of the last four instructions contains an indirect address, referring to a field in the input area. Refer to the Program Card Format for an explanation of each instruction. The program begins at 00000, and continues:

Location	Mnemonic OP Code	Instruction	Explanation
00000	RNCD	36 19901 00500	Read card to input area
00012	TD	25 00080 1991$\overline{0}$	Save digit where record mark will fall
00024	TR	31 1990$\overline{5}$ 19920	Store the instructions from the card
00036	TD	25 1991$\overline{0}$ 00080	Put the digit back again
00048	B	49 1991$\overline{5}$ ‡	Branch to 00000 or the program

Program Card Format

19901 ↓	19910 ↓			
xxxxx	xxxxx	xxxxx	x - - - - - - - - - - x	‡
1 - 5	6 - 10	11 - 15	20 - 79	80
Address for card col. 20	Address where ‡ will fall	Blank except last card	Program instructions or constants. Last digit is followed by a ‡	

Card Columns	Explanation
1-5	The address to which the data from card column 20 and on, will be transferred. It is referred to indirectly in the third instruction. No flag is needed with this or the other addresses.
6-10	The address to which the record mark will be transferred by the third instruction. This address is referred to indirectly to save and later to restore the digit that will be erased by the record mark. If the digit at that address will be loaded by a card that follows, and if proper program card sequence can always be maintained, the address may be left blank.
11-15	Blank in all but the last program load card to cause a branch to 00000. In the last program load card, these columns contain the address at which the main program is to start.

16-19 Not used. Card sequence number or program number may be punched here.

20-80 The data to be loaded, followed by a record mark. The data may be any numerical data, flagged or not. It may not contain a record mark. The first character is in card column 20. If one instruction is to be loaded, the record mark should be in column 32.

Load Routine Analysis

The first instruction, 36 19901 00500, reads the data from each card into the load area (19901-19980 in this case). A record mark is placed in 19980.

The second instruction, 25 00080 1991$\bar{0}$, transfers the digit that the record mark in 19980 will replace during the third instruction to 00080. Note the indirect Q address (1991$\bar{0}$) — the field at 19910 was loaded by columns 6-10 of the card. For example, if the data in 19901-19980 is to be transferred to 10000-10079 by the third instruction, the digit at 10079 is transferred to 00080 by the second instruction. The fourth instruction returns the saved digit to 10079.

The third instruction, 31 1990$\bar{5}$ 19920, transmits the five instructions and the record mark from 19920-19980 to the address specified by the data at 19905 (1990$\bar{5}$ is an indirect address). This address was punched in columns 1-5 of the card.

The fourth instruction, 25 1991$\bar{0}$ 00080, transfers the saved digit back to its original position, and wipes out the record mark.

The fifth instruction, 49 1991$\bar{5}$, branches the computer back to 00000, and the next card is read. The P address is indirect (19915) — the field at 19915 was loaded by card columns 11-15. These columns are blank for all but the last card. Blank card columns read into core storage as zeros.

Program Card Examples

1-5	6-10	11-15	16-19	20-79	80
00600	00660	Blank	1	Five instructions	‡
00660	00720	Blank	2	Five instructions	‡
00100	00160	Blank	3	Data for locations 100-159	‡
00340	00401	Blank	4	Data for locations 340-399	‡

Cards 1 and 2 are standard 5-per-card load cards. Card 3 loads 60 digits of the multiply table. Card 4 loads the last 60 digits of the addition table and also loads a record mark to 00400.

Loading Record Marks

1-5	6-10	11-15	16-19	20-79	80
00800	00860	Blank	5	Five instructions	‡
00811	00812	Blank	6	‡ followed by blanks	Blank

Cards 5 and 6 show a method of loading a record mark between instructions. If a record mark is to be loaded to location 00811, the record mark in column 31 of card 5 must be omitted because it would terminate the third load instruction too soon. The record mark is loaded by card 6. The instructions from card 5 are loaded into 00800-00859.

Patch Cards

1-5	6-10	11-15	16-19	20-31	32	33-80
00600	00612	Blank	7	A single instruction	‡	Blank
00700	00712	Blank	8	Instruction with ‡ in Q_{11}	Blank	Blank

Cards 7 and 8 load single instructions. If there were patches to the program in cards 1 and 2, they would have to follow cards 1 and 2 in the program deck, though not necessarily directly behind cards 1 and 2.

Last Program Load Card

1-5	6-10	11-15	16-19	20-79	80
19000	19060	00500	9	Five instructions	‡

In this example, the loader stores the instructions in card 9 to locations 19000-19059 and then branches to 00500 to start the program. A card like this is the last card of the program deck. <u>It should follow all patch cards.</u>

Program Initialization

To load and execute an initialization program (defining fields, establishing constants, etc.) before loading the rest of the program, punch the starting address of the initialization program in columns 11-15 of its last program load card. Do not alter the loader (locations 00000-00060) while initializing. After initialization, branch to 00000 and loading will resume. This allows the initialization program deck and the main program deck to be loaded as a single deck by a common program loader.

Alternative Read-in Area

To use locations 00081-00160 as a program read-in area, the program in the loader should be changed to:

36 00081 00500 25 00060 0009$\bar{0}$ 31 0008$\bar{5}$
00100 25 0009$\bar{0}$ 00060 49 0009$\bar{5}$

In the last program load card, columns 20-79 must contain the multiply table digits for locations 00100-00159, and columns 1-15 must contain 00100 00160 xxxxx, where xxxxx is the address at which the program is to start.

Mode	Character	Input			Core Storage		Output		
		Typewriter	Tape	Card	Alpha	Num	Typewriter	Tape	Card
ALPHAMERIC MODE	(Blank)	(Space)	C	(Blank)	C	C	(Space)	C	(Blank)
	. (Period)	.	X0821	12,3,8	C	3	.	X0821	12,3,8
	)	)	X0C84	12,4,8	C	4	)	X0C84	12,4,8
	+	+	X0C	12	1	C	+	X0C	12
	$	$	XC821	11,3,8	1	3	$	XC821	11,3,8
	*	*	X84	11,8,4	1	4	*	X84	11,4,8
	- (Hyphen)	-	X	11	2	C	-	X	11
	/	/	0C1	0,1	2	1	/	0C1	0,1
	, (Comma)	,	0C821	0,3,8	2	3	,	0C821	0,3,8
	(	(	084	0,4,8	2	4	(	084	0,4,8
	=	=	821	3,8	3	3	=	821	3,8
	@	@	C84	4,8	3	4	@	C84	4,8
	A-I	A-I	X0, 1-9	12, 1-9	4	1-9	A-I	X0, 1-9	12, 1-9
	0 (—)	(None)	(None)	11,0	5	C	- (Hyphen)	X	11,0
	J-R	J-R	X, 1-9	11, 1-9	5	1-9	J-R	X, 1-9	11, 1-9
	1-9 (—)	J-R	X, 1-9	11, 1-9	5	1-9	J-R	X, 1-9	11, 1-9
	S-Z	S-Z	0, 2-9	0, 2-9	6	2-9	S-Z	0, 2-9	0, 2-9
	0 (+)	0	0	0 or 12,0	7	C	0	0	0
	1-9 (+)	1-9	1-9	1-9	7	1-9	1-9	1-9	1-9
	⧧	⧧	082	0,2,8	C	C82	(Stop)	EOL	0,2,8
NUMERICAL MODE	(Blank)	(Space)	C	(Blank)		C	0	0	0
	0 (+)	0	0	0		C	0	0	0
	0 (-)	0̄	X, X0C	11,0		F	0	X	11,0
	1-9 (+)	1-9	1-9	1-9		1-9	1-9	1-9	1-9
	1-9 (-)	1̄-9̄	X, 1-9	11, 1-9		F, 1-9	1-9	X, 1-9	11, 1-9
	⧧	⧧	082	0,2,8		C82	(Stop, WN) ‡ (DN)	EOL(WN) 082 (DN)	0,2,8
	⧧̄	⧧̄	X82	11,8,2		F82	⧧̄	X82	11,8,2
	⯒	⯒	C8421	0,7,8		*C8421	⯒	08421	0,7,8
	⯒̄	⯒̄	CX08421	12,7,8		F8421	⯒̄	CX08421	12,7,8
	Num Blank †	@	C84	4,8		C84	@	C84	(Blank)

† For Card Format Use Only
* Recorded as 0, 8, 4, 2, 1 in disk storage

APPENDIX B. TAPE SPLICING

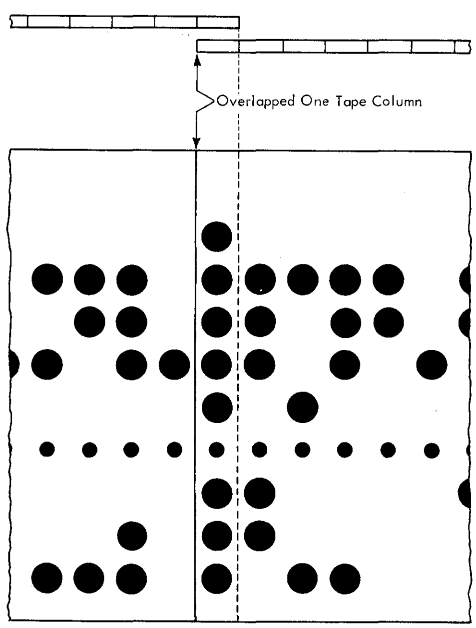
The use of tape splicing equipment should be considered if it becomes necessary to repeatedly edit tape or alter the length of tape. Special splicing equipment and materials will provide efficient, accurate, and more permanent tape splices. The selection of appropriate splicing equipment, from the many types now being offered by various manufacturers depends upon the quality of the splice desired, life expectancy of the splice, time allotted to make the splice, and a price justification. The best splicing results will be obtained by first analyzing paper tape splicing needs and then purchasing the tape splicer and splicing materials best suited to these needs.

TYPES OF TAPE SPLICES

There are two types of splices, overlap splice and butt-joint splice.

Overlap Splice

The overlap splice consists of two matching paper tape ends overlapped by at least one tape column and held together with an adhesive.



With some splicing equipment, the pieces are welded together through a process of heat, pressure, and a liquid bonding agent. With this type of equipment, alignment accuracy of the tape is not required of the splice equipment operator, and the tape splicing rate is approximately one per minute.

With other overlap-type splice equipment, the tapes are glued together with a quick drying adhesive. In this process, some alignment accuracy of the tapes is required of the splice equipment operator. Tape splicing rate is about three to five minutes per splice.

Advantages of Overlap Splicing

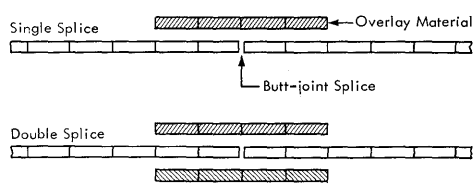
1. A large variety of overlap splicing equipment is commercially available.
2. Many splicers are available at a low investment cost.
3. Quality of splice is not usually dependent upon the operator's skill.

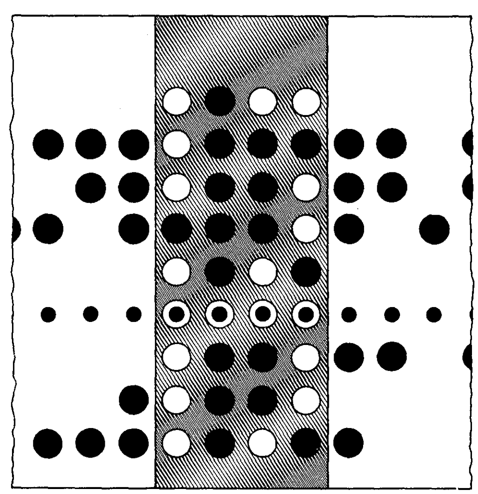
Disadvantages of Overlap Splicing

1. An overlap-type of splice is not suitable within the tape data area because overlapped columns will result in lost data, parity errors, and/or invalid codes.
2. Low production on some splicing equipment, due to the lengthy time required for the glue to dry.
3. Short life of splice.

Butt-Joint Splice

The butt-joint splice consists of two symmetrically matched paper tape ends, butted together and held into position by a bonding agent and an overlay material. The overlay material can be plastic or paper, and can be placed on one or both sides of the tape.





With some splicing equipment, the overlay material is heat sensitive and is bonded to the tape through the use of a heated iron. Alignment accuracy of the tapes is not required of the tape splicing operator. Tape splicing rate is approximately one per minute.

With other butt-joint splice equipment, the paper is bonded to the overlay material by an adhesive on the overlay material. Alignment accuracy of the tapes depends upon the skill of the splice equipment operator. Tape splicing rate is approximately one per minute.

Advantages of Butt-Joint Splice

1. The splice can be made in a data-portion of tape without losing or altering data.
2. This type of splice permits tape repairing due to tears or damage.
3. A thinner splice — the total thickness of the tape and splice (with a plastic overlay) is usually thinner than two thicknesses of paper tape.
4. Reasonably long life of splice.

Disadvantages of Butt-Joint Splice

1. The quality and accuracy of each splice may depend upon the skill of the splice equipment operator.
2. There is a limited variety of accurate butt-joint splicing equipment.

It is possible to read data recorded on IBM cards in Binary-Coded Decimal (BCD) form directly into the 1620 Data Processing System or 1710 Control System from the 1622 without first transforming the data to decimal codes.

The Binary-Coded Decimal codes shown in Table 1, are translated without read checks or parity checks. However, if the 12 punch is used for positive indication, then the 8, 2, 12 combination results in a flagged record mark ($\bar{‡}$). All other BCD card punches result in core storage values independent of a 12 punch, i.e., 4, 1, 12 gives 5.

When a card is read by the 1622, the data is stored in the buffer in BCD notation. For example, if a 1 and 8 are punched in the same column of the card, they are stored in the buffer as a 1 and 8, which is the BCD coding for a 9. (Note that a 9 punch also produces a 1 and 8 code in the buffer.) Zone punches in the card are also converted to the appropriate BCD coding. Thus an X, 1, 4 punched in a column is stored as an "N" in BCD. BCD buffer codes are shown in Table 2.

The data transferred to core storage is under control of the Read instruction. A "Read Numerically" instruction reads the X, 1, 4 into core storage as a flagged five ($\bar{5}$). A "Read Alphamerically" instruction reads the X, 1, 4 into core storage as a fifty-five (55).

Table 1. Valid Codes with Read Numerically

Positive Numbers		Negative Numbers	
Card Punches	Core Storage Value	Card Punches	Core Storage Value
1	1	1, 11 (or X punch)	$\bar{1}$
2	2	2, 11	$\bar{2}$
2, 1	3	1, 2, 11	$\bar{3}$
4	4	4, 11	$\bar{4}$
4, 1	5	4, 1, 11	$\bar{5}$
4, 2	6	4, 2, 11	$\bar{6}$
4, 2, 1	7	4, 2, 1, 11	$\bar{7}$
8	8	8, 11	$\bar{8}$
8, 1	9	8, 1, 11	$\bar{9}$
8, 2	‡ (RM)	8, 2, 11	$\bar{‡}$ (RM)

Table 2. Character Coding Chart

Character	Card	Read Buffer
Blank	Blank	C
.	12,3,8	XO821
)	12,4,8	CXO84
-	11	X
$	11,3,8	CX821
*	11,4,8	X84
,	0,3,8	CO821
(	0,4,8	O84
+	12	XOC
=	3,8	821
@	4,8	C84
A-I	12,(1-9)	XO,(1-9)
J-R	11,(1-9)	X,(1-9)
/	0,1	CO1
S-Z	0,(2-9)	O,(2-9)
0-9(+)	(0-9)	(0-9)
$\bar{1}$-$\bar{9}$(-)	11,(1-9)	X(1-9)
$\bar{0}$(-0)	11,0	CX841
0(+0)	12,0	O
‡	0,2,8	O28
$\bar{‡}$	11,2,8	X28
Num. Blank	4,8	C,8,4
⧧	0,7,8	O8421
$\bar{⧧}$	12,7,8	CXO8421

INDEX

A26-5707-0

IBM 1620/1710 Printed in U. S. A. A26-5707-0

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, New York